

Robust Object Tracking System Based on Face Detection

Min Seok Kwak[†]

ABSTRACT

Embedded devices with the development of modern computer technology also began equipped with a variety of functions. In this study, to provide a method of tracking efficient face with a small instrument of resources, such as built-in equipment that uses an image sensor in recent years has been actively carried out. It uses a face detection method using the features of the MB-LBP in order to obtain an accurate face, specify the region (Region of Interest) around the face when the face detection for the face object tracking in the next video did. And in the video can not be detected faces, to track objects using the CAM-Shift key is a conventional object tracking method, which make it possible to retain the information without loss of object information. In this study, through the comparison with the previous studies, it was confirmed the precision and high-speed performance of the object tracking system.

Keywords : Embedded Devices, CAM-Shift, MB-LBP, ROI, Face Detection

얼굴검출에 기반한 강인한 객체 추적 시스템

곽민석[†]

요약

최근 컴퓨터 기술의 발전과 함께 임베디드 기기 또한 다양한 기능을 갖추기 시작했다. 본 연구에서는 최근 활발하게 진행되고 있는 영상센서를 사용한 임베디드 기기 등 자원이 적은 기기에서 효율적인 얼굴 추적 방식을 제안한다. 정확한 얼굴을 얻기 위하여 MB-LBP 특징을 사용한 얼굴 검출 방식을 사용했으며, 다음 영상에서 얼굴 객체 추적을 위하여 얼굴 검출시 얼굴 주변 영역(Region of Interest)을 지정하였다. 그리고 얼굴을 검출을 못하는 영상에서는 기존의 객체 추적 방식인 CAM-Shift를 사용해 객체를 추적해 객체 정보의 손실 없이 정보를 유지할 수 있도록 하였다. 본 연구는 기존 연구와의 비교를 통하여 객체 추적 시스템의 정확성과 빠른 성능을 확인하였다.

키워드 : 임베디드 장치, CAM-Shift, MB-LBP, 관심영역, 얼굴 검출

1. 서론

최근 사물인터넷의 유행으로 임베디드 기기 또한 각광을 받고 있다. 임베디드 기기의 발전과 함께 영상 센서를 사용한 영상 처리 분야 또한 주목을 받고 있다. 임베디드 기기를 사용한 영상 처리 분야는 자동차, 사물인터넷, 보안 등의 폭 넓은 분야에서 사용을 한다. 하지만 많은 자원을 요구하는 영상 처리 기술은 임베디드 기기에서 사용상 많은 제약이 따를 수밖에 없다. 본 연구에서는 제한된 자원을 효율적으로 사용할 수 있는 방법을 제안한다. 전통적인 얼굴 검출 방법과 객체 추적 방법을 사용해 얼굴을 검출하여 추적 한

다. Viola-Jones가 제안한 Adaboost를 사용한 얼굴 검출 방식은 전통적인 얼굴 검출 알고리즘으로 얼굴에서 학습되어 있는 정보를 사용해 정확한 얼굴 검출이 가능하지만 다양한 각도의 얼굴 객체를 검출하기 힘들며, 기본적으로 기계 학습 알고리즘은 자원을 많이 사용해 객체 추적 방식에는 적합하지 않다[2]. 반면 전통적인 객체 추적 방식인 CAM-Shift 방식은 색상 정보를 기반으로 정확한 얼굴을 추적하기는 힘들지만 다양한 각도 변화에서 강인한 객체 추적을 할 수 있다[4]. 본 연구에서는 전통적인 얼굴 검출 기법과 객체 추적 기법을 사용해 각각의 알고리즘을 효율적으로 사용할 수 있는 객체 추적 시스템을 제안한다.

본 연구는 먼저 연구에서 사용한 MB-LBP알고리즘과 CAM-Shift알고리즘을 설명한다[4]. 다음 장에서는 본 연구에서 제안하는 객체추적방식을 설명한다. 그 후 기존 알고리즘과의 비교를 통한 성능 비교를 한 후 결론으로 끝이 난다. 본 논문은 2016년 춘계학술발표대회에서 발표한 논문을 확장한 논문이다.

* 이 논문은 2016년도 한국정보처리학회 춘계학술발표대회에서 '얼굴 검출에 기반한 강인한 객체 추적 시스템'의 제목으로 발표된 논문을 확장한 것임.

[†] 준회원: 한국디지털미디어고등학교 학생

Manuscript Received: July 1, 2016

First Revision: August 2, 2016

Accepted: August 3, 2016

* Corresponding Author: Min Seok Kwak(kms7530@gmail.com)

2. MB-LBP (Multi-Scale Block LBP)

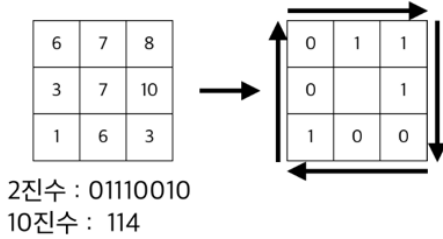


Fig. 1. The operation of the LBP

LBP(Local Binary Pattern)는 주로 영상에서는 효과를 주기 위해 만든 필터로 주로 사용된다. 최근 LBP는 이용하여 얼굴 검출에도 널리 사용하며 얼굴 외에도 객체 검출에서도 강한 성능을 보여준다. 기존 LBP는 3×3인 블록에서 중심에 있는 픽셀을 기준으로 주변의 화소 값이 크거나 같으면 1, 작으면 0을 넣는다. 그 후 각 픽셀의 값들을 취합해 2진수의 수를 10진수로 나타낸다. 일반적인 Equation (1), Equation (2)와 같다[13].

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c)2^p \quad (1)$$

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (2)$$

이후 LBP로 변환한 이미지의 히스토그램을 이용하여 얼굴인식을 하게 된다. 아래의 사진은 LBP적용 후 히스토그램으로 표현한 것이다.



Fig. 2. LBP Processing and Color Information for Face Recognition

그러나 위의 Fig. 2와 같이 기존의 LBP를 적용한 히스토그램의 경우 픽셀 단위로 작업하기 때문에 국부적인 특징만 찾아낼 수 있다. 이 문제점을 해결하기 위해 나온 것이 2007

년에 발표된 MB-LBP (Multi-Scale Block LBP)이다. MB-LBP는 3×3의 제한된 픽셀이 아닌 임의의 크기의 영상영역에서 3×3의 블록으로 나누어 그 블록의 중심을 기준으로 계산하기 때문에 국부적으로 특징을 찾아내는 기존LBP와 비교해 큰 스케일의 텍스처 정보를 추출할 수 있다. 즉 픽셀 단위로 특징을 찾게 되는 LBP를 확장하여 MB-LBP는 픽셀이 아닌 블록단위의 연산을 하여 특징을 찾기 때문에 한 대상에 대해서 다양한 특징을 찾을 수 있게 된다.

3. CAM-Shift (Continuously Adaptive Mean-Shift)

CAM-Shift는 크기의 유동성을 스스로 조절하는 기법을 사용하여 추적하는 객체의 크기가 변하며 색상 정보가 변할 경우 찾아내지 못하는 Mean-Shift의 단점을 보완하여 나온 알고리즘이다. 기초적인 추적 방식은 색상 정보를 사용하기 때문에 객체 추적을 위하여 먼저 지정 영역 주변의 추적할 객체의 관심영역이 있어야 한다. 여기서 고려하게 되는 요소는 초기 위치, 객체의 대칭 또는 비틀림, 회전, 원형 또는 사각형 그리고 크기를 고려한다. 그 후 관심 영역을 찾은 뒤 그 관심영역의 HSV정보를 색상정보로 변환한다. 그리고 HSV로 변환한 영상을 색상 정보로 변환해 객체 추적 정보로 사용한다. 그리고 탐색 구간을 중심으로 객체를 탐색하는데 관심영역으로 지정된 곳과 비슷한 색상 정보를 찾게 된다. 그 후 색상 정보 변환을 한 영역에서 지정 영역의 색상 정보와 비교하여 물체를 추적한다. 이 추적방식의 과정에서는 해당 영역에서 비슷한 영역을 가중치를 더 주게 되는데 추적하는 동안 이 작업을 계속하여 이동 벡터가 0이 될 때까지 이동한다. 그 후 이동 벡터가 0에 수렴하게 되는 경우 탐색을 멈추고 결과를 출력한다. 아래는 CAM-Shift가 동작하는 방식에 대해 도식화해 놓은 Fig. 3과 CAM-Shift의 수학적 Equation (3)이다.

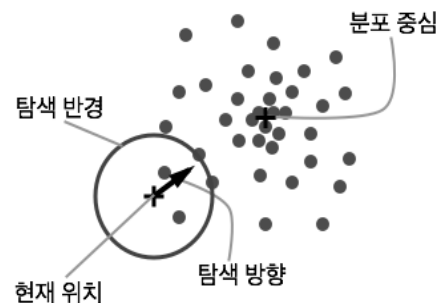


Fig. 3. Schematic of the Operation of the CAM-Shift

$$\frac{\sum_{i=1}^{n_h} X_i \omega_i g \left(\left\| \frac{\hat{y}_0 - X_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} \omega_i g \left(\left\| \frac{\hat{y}_0 - X_i}{h} \right\|^2 \right)} \quad (3)$$

$$f(k+1) = \begin{cases} \text{if } x > 0 & \sum_{p=0}^{P-1} s(i_p - i_c) 2^p \\ \text{if } x = 0 & \frac{\sum_{i=1}^{n_h} X_i \omega_i g \left(\left\| \frac{\hat{y}_0 - X_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} \omega_i g \left(\left\| \frac{\hat{y}_0 - X_i}{h} \right\|^2 \right)} \end{cases} \quad (4)$$

4. 얼굴 검출에 기반한 객체 추적 시스템

처음의 영상이 들어오게 되면 그 영상에서 지정 영역을 사용하여 MB-LBP 얼굴 검출을 시도한다. 얼굴 검출에 성공하면 검출된 얼굴 정보를 추적 정보로 사용해 검출 영역을 지정한다. 여기에서 지정 영역을 사용하는 이유는 한 영상에서 검색 시도 횟수를 줄이기 위해서 얼굴 검출이 될 가능성이 높은 부분에서만 검색하기 위해서다. 검출 영역 지정 작업을 통해서 객체 검출 시 영상의 모든 부분에서 검출을 시도하는 것보다 높은 성능을 낼 수 있다. 그러나 만약 얼굴 검출을 시도했으나 검출되지 않으면 객체 추적을 시도한다. 추적을 시도할 때는 전 영상에서 저장된 얼굴 정보를 사용하여 추적하게 된다. 얼굴 추적의 경우 또한 찾아낸 얼굴 정보를 새로 갱신하여 다음 영상에서 객체를 찾는데 사용한다. Fig. 4는 전체 흐름도를, Equation (4)는 추적시스템을 나타낸 것이다.

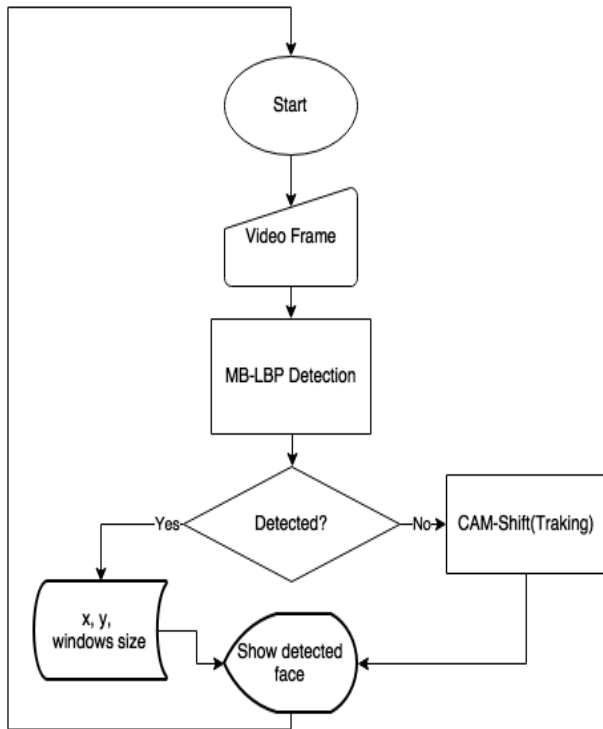


Fig. 4. Full Flow Chart to Suggest Tracker

k는 해당 프레임의 번호, x는 검출된 얼굴의 개수이다. 초기 f(0)의 상태에서는 초기에 검출된 얼굴 정보가 없으므로 모든 프레임에서 얼굴 검출을 시도한다. 그 후 검출된 얼굴 정보(얼굴 위치, 얼굴 색상)를 기준으로 다음 프레임에서 얼굴을 찾게 된다. 다음 프레임에서 얼굴 검출 시 얼굴 정보를 통해 ROI영역을 지정해 얼굴 검출시도 시 얼굴 개수가 0보다 크면 얼굴 검출로 검출된 얼굴 개수가 0보다 작을시 객체 추적을 시도한다.

5. 실험 결과

Fig. 4의 위의 행과 같이 영상에서 겹침이 발생한 경우 이번에는 객체를 찾지 못한다. 이 경우 객체 추적 도중 실패하게 되지만 그러나 본 연구에서 제시하는 방식은 Fig. 4와 같이 얼굴 검출이 불가능 한 구간에서는 객체 추적을 하게 된다. Fig. 5의 위의 행은 객체 추적이다. 객체 추적의 원래 문제점은 객체 검출보다 정확성이 떨어지는 단점이 있는데 반해, 제안하는 추적기는 객체 검출이 가능한 구간에서는 객체 검출을 시도를 함으로서 객체추적보다는 좀 더 정밀한 검색을 시도 할 수 있게 된다. 다음 실험데이터는 본 연구에서 제안한 물체 추적 방법과 기존의 방식인 객체 추적과 객체 인식과의 성능을 비교이다. 첫 번째 Fig. 7은 객체 검출, 객체 추적, 제안하는 추적기의 속도를 비교하였다.



Fig. 5. An Existing Object Detection and Proposed Tracker



Fig. 6. An Existing Object Tracking and Proposed Tracker

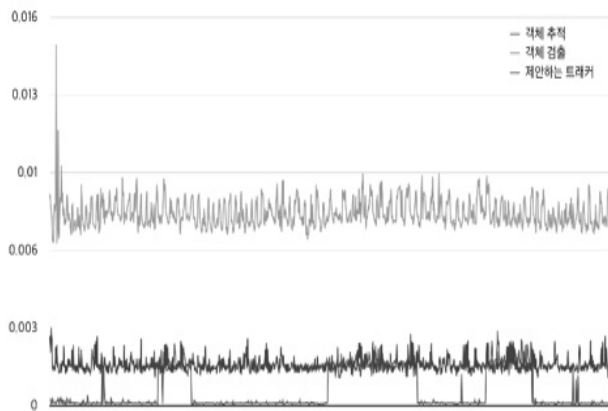


Fig. 7. Speed Comparison Graph

초록색선인 객체 검출은 가장 높은 시간을 보여줬다. 또한 파란색선인 객체 추적은 제안한 추적기 방식보다 좀 더 높은 시간을 보여줬다. 얼굴 검출은 7.10초로 가장 많은 시간이 걸렸고, 다음으로 객체 추적은 1.52초 그리고 마지막으로 제안한 추적기는 0.57초로 최단 시간을 보여줬다. 이는 객체 검출 대비 최대 80.74%의 개선된 속도를 보여줬다.

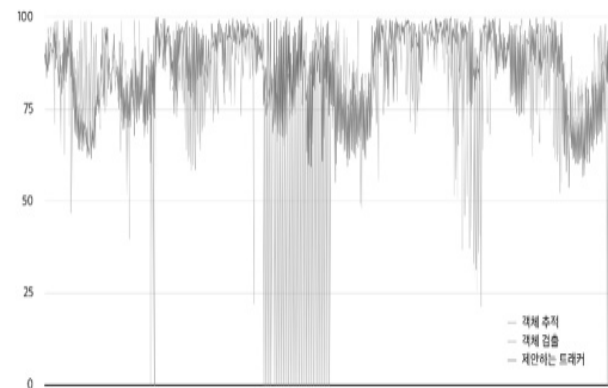


Fig. 8. Accuracy Comparison Graph

두 번째 Fig. 8에서는 정확도를 비교 하였다. 객체 검출인 초록색 그래프는 그래프와 같이 겹침(Occlusion)이 발생한 구간에서는 찾지 못하여 정확도가 0%인 것을 확인할 수 있다. 또한 객체 추적인 파란색 그래프에서는 특정 구간에서는 객체 검출보다 뛰어난 성능을 보여주지만 객체를 찾던 도중 다른 곳을 찾아 버리는 일이 발생해 특정 상황에서 좋지 않은 결과를 볼 수 있다. 그러나 위의 두 경우에 반해 제안한 추적기는 객체 검출을 먼저 시도해, 보다 정확한 위치를 찾기 위해 시도한다. 그 후 겹침, 폐색 등 특정 구간에서 객체를 찾지 못하는 경우 전 영상에서 저장된 얼굴 데이터를 이용하여 객체 추적을 시도한다. 위의 그래프에서 나오는 것과 같이 제안하는 추적기는 빨간색 그래프이다. 빨간색 그래프는 위의 두 경우와 다르게 좀 더 높은 정확도를 갖고 있는 것을 확인할 수 있었다.

또한 실험 데이터뿐만 아니라 실제 상황에서도 사용이 가능한지 노트북의 웹캠으로 실험해 보았다. 실험데이터는 아래의 Fig. 9와 같다.

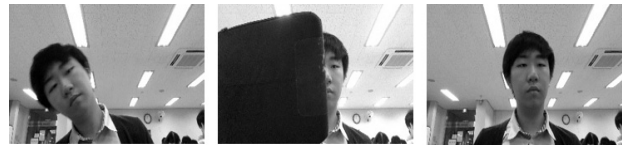


Fig. 9. Actual Experimental Data

위의 사진과 같이 실험에 쓰인 데이터는 겹침 현상과 기울어짐이 있는 데이터로 실험을 해보았다. 본 데이터들은 320×180로 총 573개의 프레임들로 구성되어 있다. 결과는 아래의 Fig. 10과 같다.

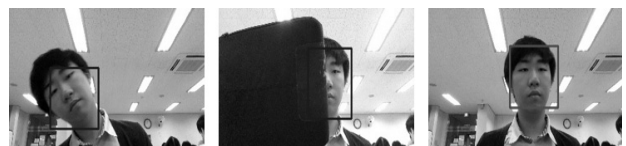


Fig. 10. The Results of the Actual Experimental Data

위의 경우와 같이 객체 검출이 원활한 경우 객체 검출을 시도해 더 정확한 결과물을 도출하고, 겹침 현상 혹은 기울어짐이 발생하여 객체 검출이 원활하지 않은 경우 기존에 저장된 위치값과 색상 히스토그램을 이용해 해당 얼굴의 위치를 찾아 결과를 도출해낸다. 그리고 객체 검출, 객체 추적 그리고 본 논문에서 제안하는 방식들과 속도를 비교해보았다. 아래는 한 프레임당 몇초가 걸리는지 나온 그래프이다.

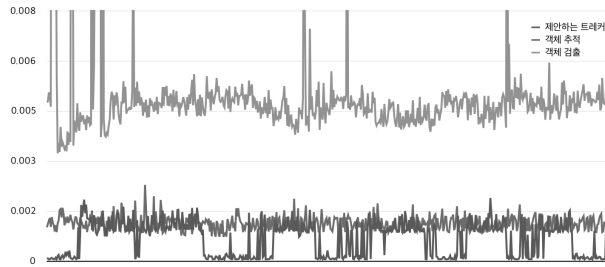


Fig. 11. Frames Per Second, Which was Extracted with Actual Data

위의 Fig. 11에서 초록색은 객체 검출, 파랑색은 객체 추적 그리고 빨간색이 본 논문에서 제안하는 방식이다. 그래프는 객체 검출시 특정 프레임이 최대 0.12로 다른 데이터가 가려져 최대 0.008로 설정한 상태이다. 객체 추적과 비교했을 때 본 논문에서 제안하는 방식이 약 0.41초로, 객체 추적이 총 0.72초가 걸린것에 비해 약 58.01%의 성능 향상을 보여줬다. 또한 객체 검출과 비교했을 때 본 논문에서 제안하는 방식이 약 0.41초로, 객체 검출이 약 3.59초로 약 116.36%의 성능개선을 보였다.

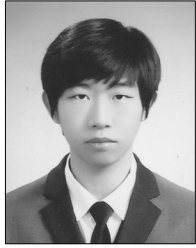
6. 결 론

최종 결론으로 제안하는 추적기는 속도와 정확도 모두 기존 연구 대비 우수한 성능을 보였다. 속도 부분에서는 기존 연구 대비 최대 80.74%의 성능 개선을 보여줬다. 또한 정확도 부분에서 최대 86.40%의 정확도를 보여 객체 추적의 85.10%, 객체 검출의 80.85%보다 높은 정확도를 보였다. 얼굴을 추적하기 매우 힘든 구간에서 객체 추적을 사용해 성능 개선을 확인하였다. 또한 실제 웹캠을 이용하여 사용하여도 객체를 추적하는데 도중 객체의 위치를 유실하는 경우를 줄이는 결과를 보였다.

마지막으로 본 논문에서 모든 실험은 ROI영역을 기존 검출위치의 110%의 영역을 탐색하는 것으로 작성되었는데 이 탐색 ROI영역을 조절하게되면 정확도를 더 높일수 있을것으로 기대된다.

References

- [1] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.24, No.1, pp. 34-58, 2002.
- [2] Paul Viola, and Michael J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, Vol.57, No.2, pp.137-154, 2004.
- [3] D. Comaniciu, V. Ramesh, and P. Andmeer, "Kernel-based object tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.25, Iss.5, pp.564-577, May., 2003.
- [4] D. Comaniciu, V. Ramesh, and P. Andmeer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.142-149, Jun., 2000.
- [5] Raul Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, Vol.57, Iss.2, pp.137-154, May., 2004.
- [6] Y. Wu, J. Lim, and M. H. Yang, "Online Object Tracking: A Benchmark," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.2411-2418, 2013.
- [7] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, Vol.29, No.1, pp.51-59, 1995.
- [8] D. P. Huttenlocher, J. J. Noh, and W. J. Rucklidge, "Tracking nonrigid objects in complex scenes," in *IEEE International Conference on Computer Vision (ICCV)*, pp.93-101, 1993.
- [9] C. Stauffer and E. Grimson, "Learning pattern of activity using real-time tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.22, No.8, pp.747-757, 2000.
- [10] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, Vol.38, No.4, pp.1-45, 2006.
- [11] T. Ojala, M. Pietikäinen, and D. Harwood, "A Comparative Study of Texture Measures with Classification Based on Feature Distributions," *Pattern Recognition*, Vol.29, No.1, pp.51-59, 1996.
- [12] T. Ahonen, A. Hadid, and M. Pietikinen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.28, Iss.12, Dec., 2006.
- [13] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li, "Learning Multi-scale Block Local Binary Patterns for Face Recognition," in *International Conference on Biometrics*, Springer Berlin Heidelberg, pp.828-837, 2007.



곽민석

e-mail : kms7530@gmail.com

2014년~현재 한국디지털미디어

고등학교 웹프로그래밍과 학생

관심분야: Computer Vision & Neural
Network